SC08445SP01
PATENT #12

gcm
7-31-97

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: : Date: July 30, 1997

Raymond H. Naugle :

Serial No.: 08/492,943 : Group Art Unit: 2302

Filed: June 21, 1995 : Examiner: M. Geckil

For: A METHOD FOR REMOTE SYSTEM PROCESS MONITORING

AFFIDAVIT UNDER 37 C.F.R. § 1.132

I, Nicholas Airdo, the undersigned, being duly sworn, depose and say:

I consider myself to be one of ordinary skill in the art of system programming and administration as evidenced by my five years of engineering practice as a system programmer and analyst. I received a B.S. degree in Computer Science from the University of Illinois in 1992.

I understand that the above identified U.S. Patent Application stands rejected under 35 U.S.C. 112, first paragraph, by the U.S. Patent and Trademark Office (USPTO) as failing to provide an enabling disclosure. I have read and understand the Office Action from the USPTO dated January 15, 1997.

I have reviewed U.S. Patent Application No. 08/492,943, entitled "A Method for Remote System Process Monitoring," by Raymond Naugle. The specification as filed adequately teaches me how to make and use the invention on or before June 21, 1995, without undue experimentation, notwithstanding the Examiner's remarks.

In general, the specification discloses a new use for a standard software program, Sendmail, which is installed on UNIX-based computers to manage email transmissions. Typically, when a monitor computer uses Sendmail to send an email message to a target computer, the email message invokes the execution of a resident program to store the message in a designated mailbox directory for later retrieval. The disclosed invention uses the ability of Sendmail to use email to invoke the execution of an alternative resident program on the target computer. More particularly, the alternative resident program is encoded to execute specific system commands other than those required to store the message in a mailbox directory. For instance, to perform a monitoring function, the resident program can execute the UNIX "ps" command such that a status table listing currently active processes on the target computer is generated. Other system commands can be included in the resident program, such as commands which format the status table as a reply message to be returned by email to the monitor computer. For such a reply message, header information in the return email similarly designates the name of the executable file on the monitor computer which is to be run when the reply message is received. As described on pages 5 and 6 of the specification, the executable file can save the status message as a file (i.e., bouncefile) in a designated directory of the monitor computer. The executable file can then invoke a polling routine to periodically poll the designated directory to determine when bouncefile is created. A static custom list of the processes expected to be active on the target computer is stored in a file in the monitor computer. When bouncefile appears, each entry of the custom list is compared with the bouncefile entries and, if an expected process is not found, an alarm is issued, such as a printout, a monitor display, a paging message, an audible alarm, a logon message, etc. For example, remote paging could be activated by calling a subroutine to dial a pager number through a modem to send an appropriate alphanumeric message or return telephone number to the pager.

The Office Action states that the specification does not teach the details of the structures or means which determine the process status. Both Sendmail and UNIX are standard programs whose operation and structures are widely used and understood. In my view, such structures and means consist of the Sendmail and UNIX software programs and their respective associated data structures. I believe such structures are adequately described on pages 5 and 6 of the specification to enable me to make and use the invention. For example, I would organize the custom list of expected active processes as a static string array. I would then compare each line of the custom list and string search each line of the status message (bouncefile) for comparing with the custom list. If a name in the custom list is not found, I would set a flag which would call the relevant subroutine to send an appropriate alarm.

The Office Action further states that the specification does not adequately teach the details of how a process which determines the process status interfaces with the received email. I respectfully disagree with the Examiner's conclusion. It is known that Sendmail can invoke the execution of a previously designated file. Such a file is designated as mbounce in the specification, which can be encoded to include the UNIX system command "ps" for monitoring the status of the target computer, as described from page 5, line 24, to page 6, line 7, of the specification. Such operation is within the capabilities of the Sendmail program and can be configured using the Sendmail companion files known as aliases.

The Examiner questions whether the specification adequately teaches the details of how a process which receives emails querying process status interfaces with another process which determines the process status. The Examiner further notes that the UNIX "ps" command can be typed on a keyboard to determine the status of active processes on a system but that the Examiner is unaware of any command that interfaces between the email and the "ps" command. It is clear that such an interface exists within the standard Sendmail program and is implemented as described above. Mbounce is encoded to execute the UNIX system command "ps" to gather status information of the target computer. Hence, system commands such as the "ps" command can be run from within mbounce when mbounce is executed in response to a properly encoded email message, rather than by typing the commands from a keyboard, as described on pages 5 and 6 of the specification.

The Office Action further questions whether the specification teaches sufficient details of how comparing the active processes on the target computer with a custom list is accomplished. I first note that the contents of a typical status message are shown on pages 6 and 7 of the Preliminary Amendment. It is evident that the status message is text-based and that the organization and syntax of bouncefile are known and are as described on page 7, lines 11-21. Therefore, I believe it would be a straightforward matter to generate a text-based string array of expected active processes which are text-formatted to be consistent with the status message. For example, I would perform a standard string compare of entries in the status message with entries in the custom list as described above. If a process in the list were not found in the status message a flag would be set which would activate the appropriate alarm.

The Office Action states that the specification fails to teach sufficient detail of how automatically activating remote paging is accomplished. I believe I would be able to provide automatically activated remote paging without undue experimentation. For example, I would activate remote paging by invoking a subroutine or program that dialed a pager number through a modem to transmit an appropriate alphanumeric message or return telephone number to the pager.

Finally, the Office Action questions whether sufficient detail has been provided in the specification to describe how heterogeneous systems interface with each other to provide the email monitoring function. The email system can be used for remote monitoring by using the capabilities of Sendmail. Therefore, the heterogeneous systems are those connected to networks running on operating systems such as UNIX that support Sendmail. Specific hardware and software interfaces between the monitor and target computers are described in the specification on page 3, lines 2-13. In particular, the heterogeneous computers run on the UNIX operating system, with email handled through the standard Sendmail program. Network communication is accomplished using the TCP/IP protocol, thereby ensuring both hardware and software compatibility on the network.

In summary, I find the specification adequate to enable me to make and use the invention as of June 21, 1995. I disagree with the rejection of the application under 35 U.S.C. 112, first paragraph, as failing to provide an enabling disclosure.

Signature of Affiant

STATE OF ARIZONA)
:
COUNTY OF MARICOPA)

I, LISA MARIE HOPKINSON, a Notary Public in and for the above County and State, certify that Nicholas Airdo, whose name is subscribed to the foregoing instrument, appeared before me this day in person and acknowledged that he signed the said instrument as his free and voluntary act for the purposes therein set forth.

Given under my hand and notarial seal this 30TH day of July, 1997.

Lisa Marie Hopkinson
Notary Public

My commission expires on June 21, 1999

OFFICIAL SEAL
LISA MARIE HOPKINSON
Notary Public - State of Arizona
MARICOPA COUNTY
My Commission Expires June 21, 1999